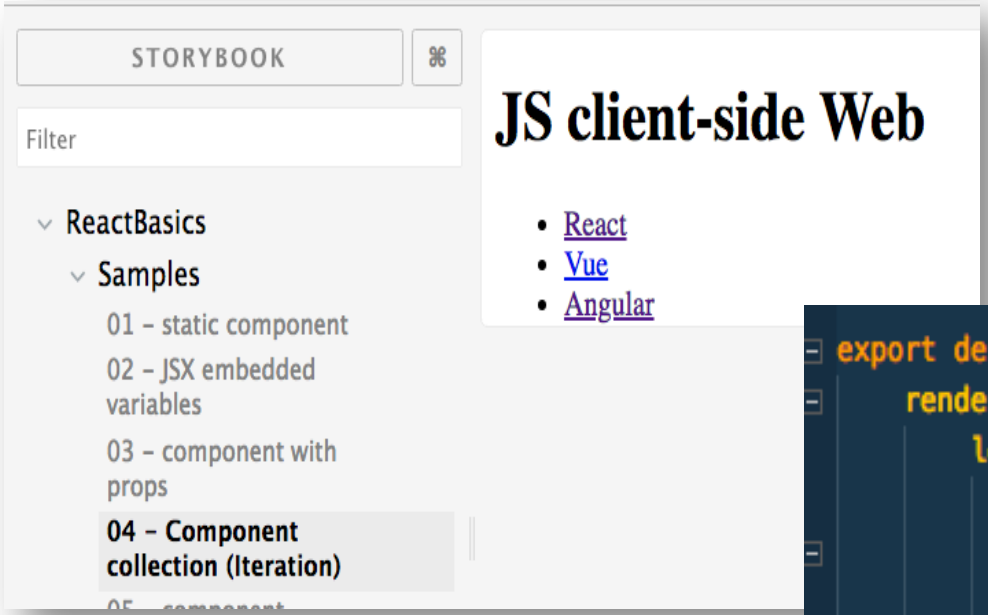
Stateless functional components

Stateless Functional components

- **Many components only require the render method.**
- **The lifecycle methods are redundant but still effect performance.**
- **Use** stateless functional components **(sfc) where possible.**

```
const ComponentName = (props) =>
    { …. body of render method …..}
```

Sample – Class component




```
.add('04 - Component collection (Iteration)',
    () => {
      let frameworks = [
        {name: 'React', url : 'https://facebook.github.io/react/'},
        {name: 'Vue', url : 'https://vuejs.org/'},
        {name: 'Angular', url : 'https://angularjs.org/'}
      ] ;
      let type = 'JS client-side Web' ;
      return  <FrameworksIteration frams={frameworks}
        type={type} /> ;
    }
  )
```

```
export default class Frameworks extends Component {
    render() {
        let list = this.props.frams.map(
            (f, index) =>
                <li key={index} >
                    <a href={f.url}> {f.name} </a>
                </li>
        );
        return (
            <div>
                <h1>{this.props.type}</h1>
                <ul>
                    {list}
                </ul>
            </div>
        );
    }
```

Sample - Stateless Functional components

```jsx
import React from 'react';

const Frameworks = props => {
    let list = props.frams.map(
        (f, index) =>
            <li key={index} >
                <a href={f.url}> {f.name} </a>
            </li>
    );
    return (
        <div>
            <h1>{props.type}</h1>
            <ul>
                {list}
            </ul>
        </div>
    );
};
export default Frameworks
```

```jsx
const Frameworks = ({frams, type} ) => {
    let list = frams.map(
        (f, index) =>
            <li key={index} >
                <a href={f.url}> {f.name} </a>
            </li>
    );
    return (
        <div>
            <h1>{type}</h1>
            <ul>
                {list}
            </ul>
        </div>
    );
};
export default Frameworks
```

Stateless Functional components

- **Legacy code.**

- **Jscodeshift – "a toolkit for running** codemods **over multiple JavaScript files".**

  $ npm install –g jscodeshift

- **react-codemods – "a collection of codemod scripts for use with JSCodeshift that help update React APIs"**

  e.g. $ jscodeshift -t ../react-codemod/transforms/pure-component.js ./components/samples/04_iteration.js --useArrows=true --destructuring=true